CASHIFY

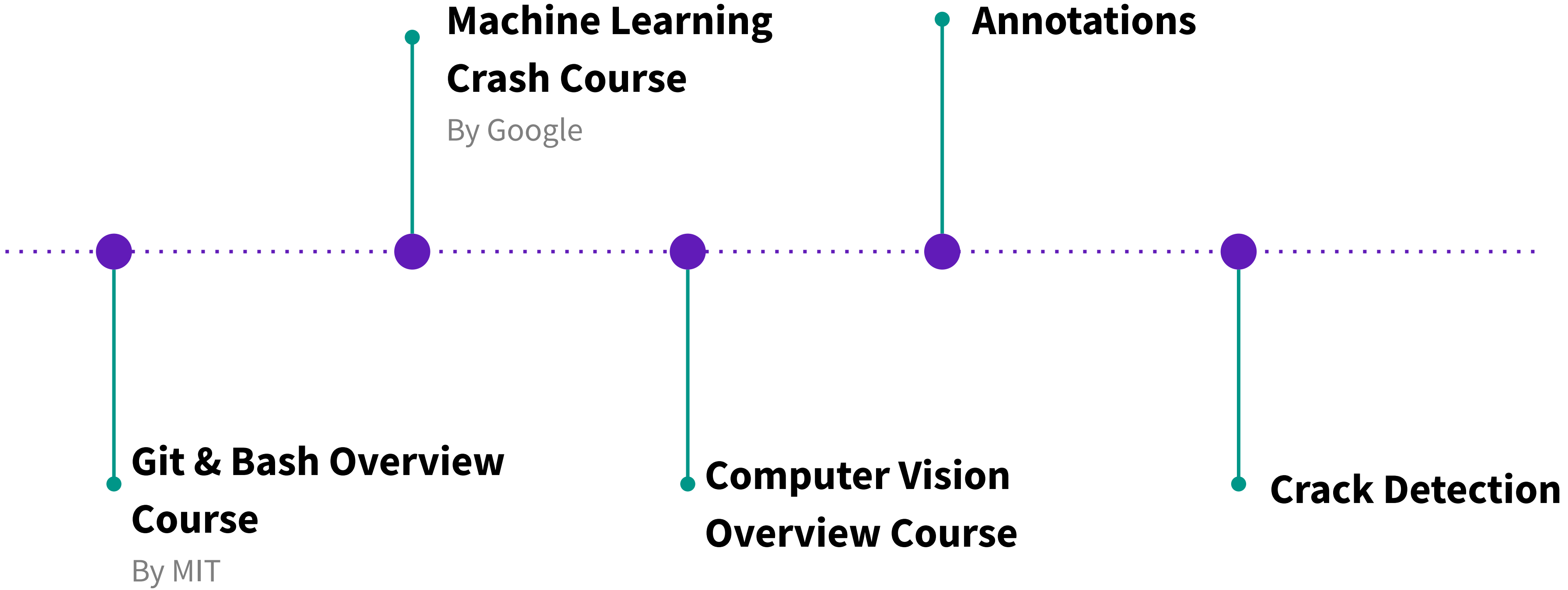
Dhruv Darda

Junior Undergrad @ IIT Gandhinagar

ML Intern
Mentor - Amit Agarwal

CASHIFY

Brief overview of the 3 month Internship



Git & Bash Overview Course

By MIT

Machine Learning Crash Course

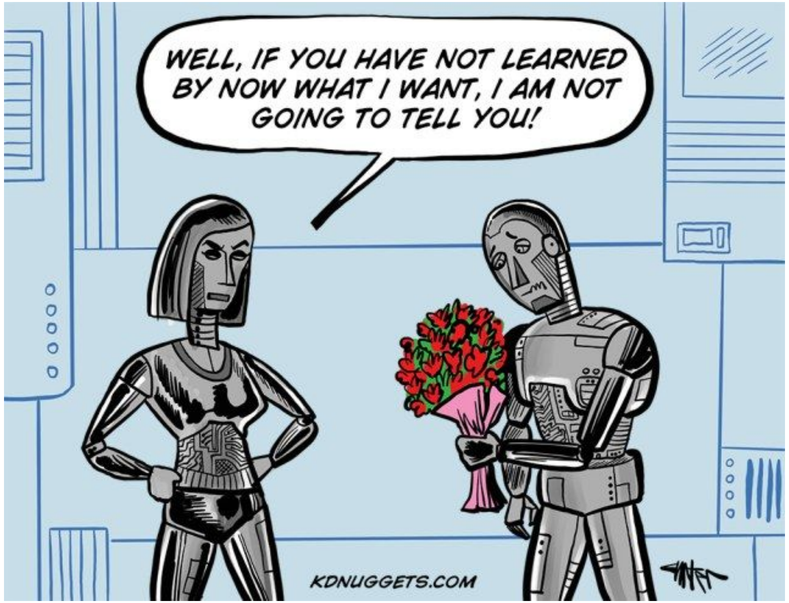
By Google

Computer Vision Overview Course

Annotations

Crack Detection

Technical Skills Learnt


WELL, IF YOU HAVE NOT LEARNED BY NOW WHAT I WANT, I AM NOT GOING TO TELL YOU!
KDNUGGETS.COM

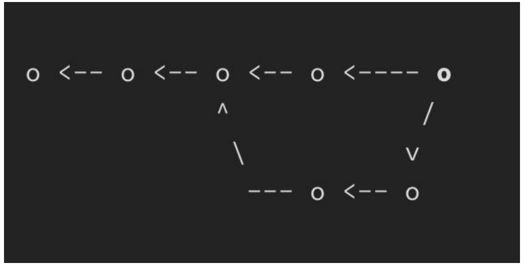
Basics of

- Computer Vision with Pytorch
- Bash
- Git

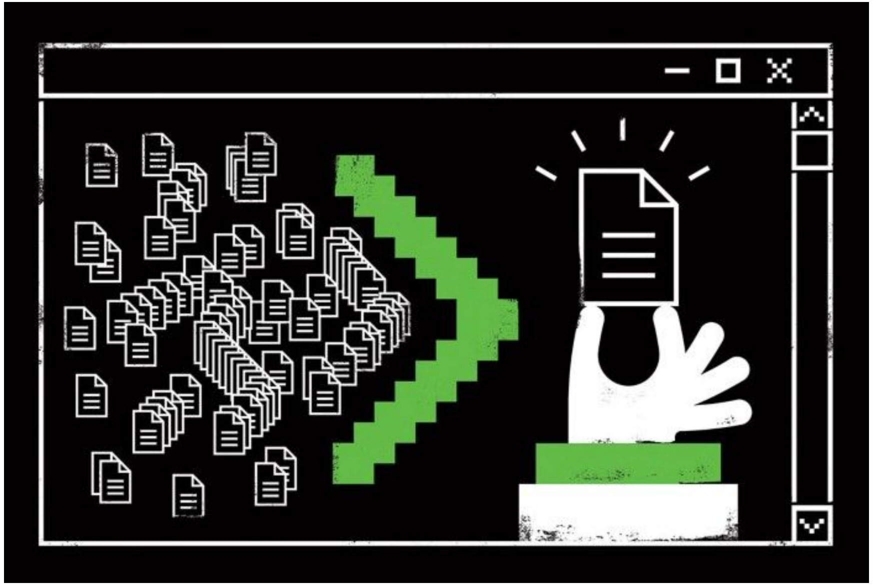
Git and Shell

- Pre-programmed commands have some limitations and we can use shell commands to tasks that are not preloaded for us.
- The basic commands like ls, cd, pwd and $PATH were used everywhere in the internship.

- Also learnt how git stores our version history in form of snapshots.



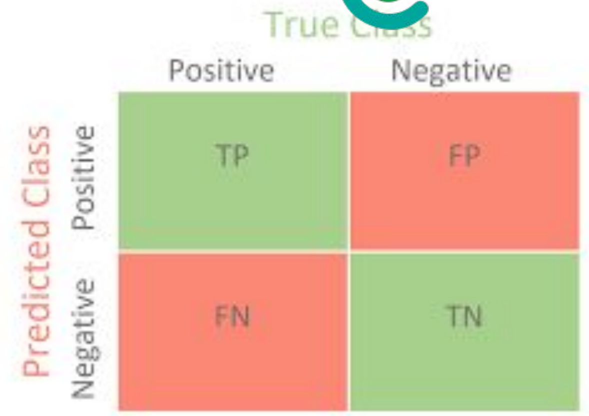
```
o <-- o <-- o <-- o <---- o
            ^            /
             \          v
              --- o <-- o
```



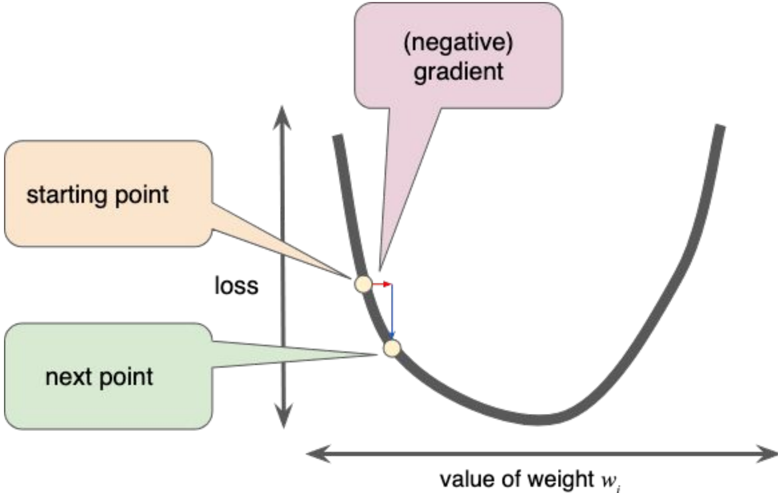
Basics of Machine Learning

Went through the Google ML Crash Course:

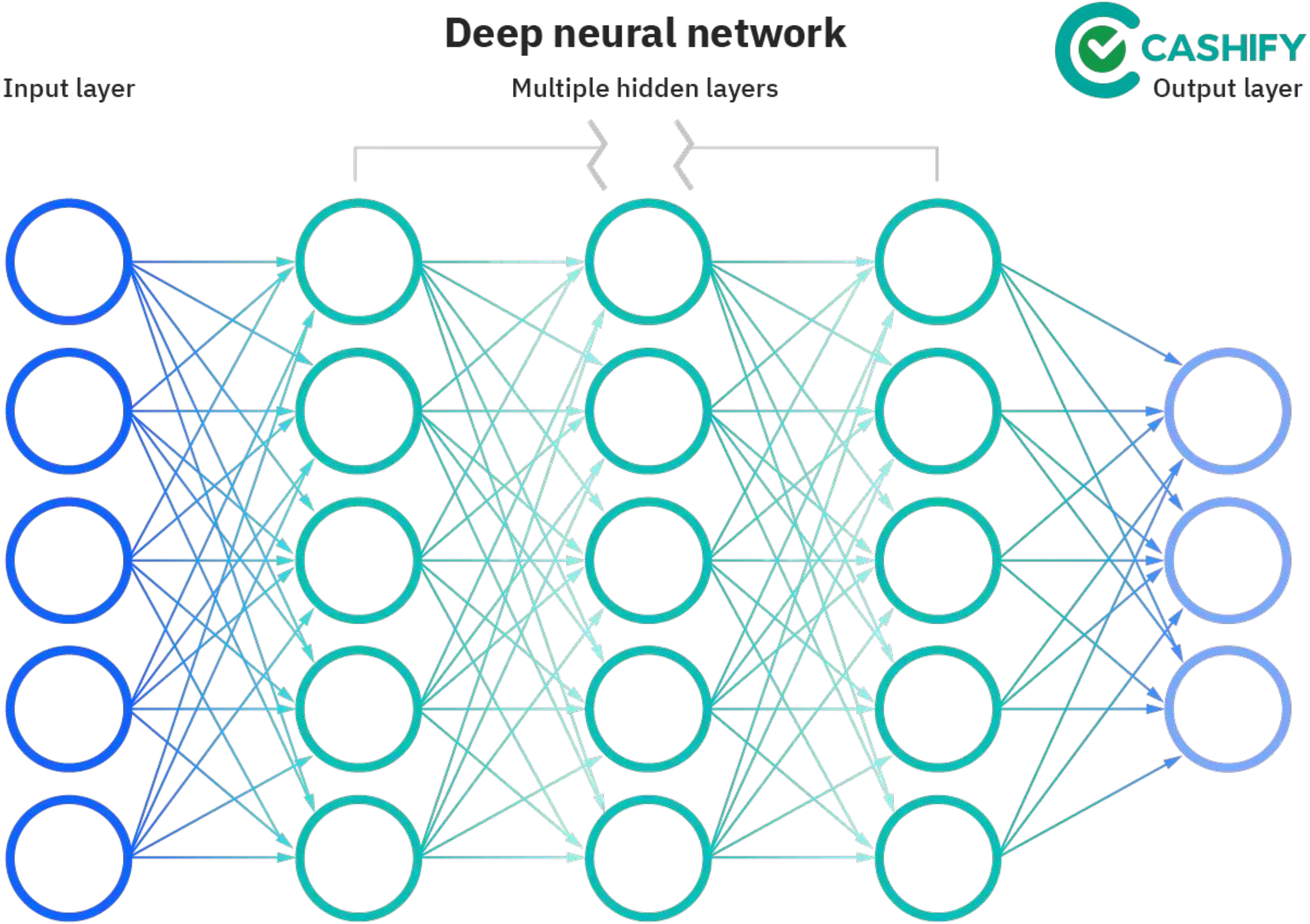
- Training and Testing
- Losses and Reduction
- Regularisation
- Regression
- Classification
- Neural Networks
- Non-Linearities



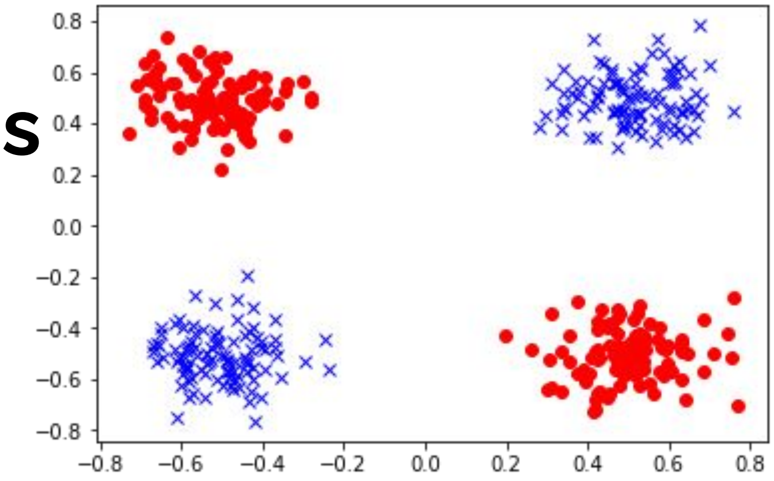
		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

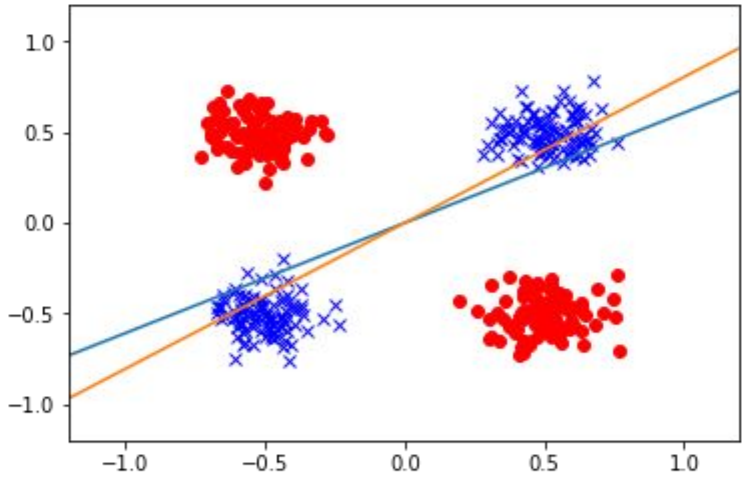


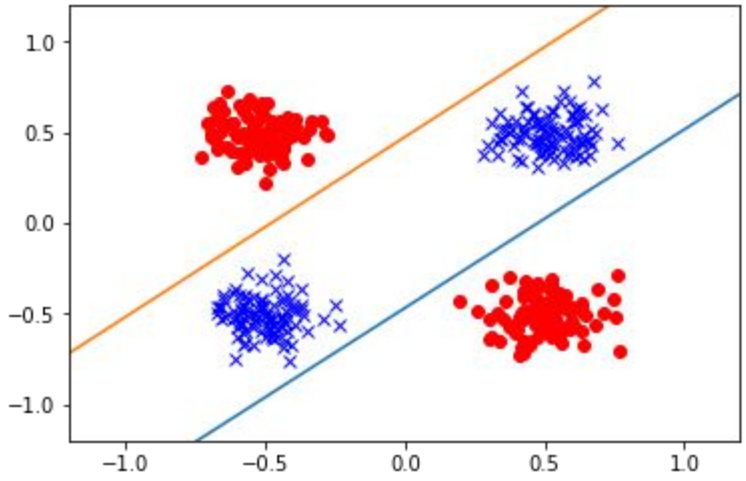
Neural Networks



Non-Linearities



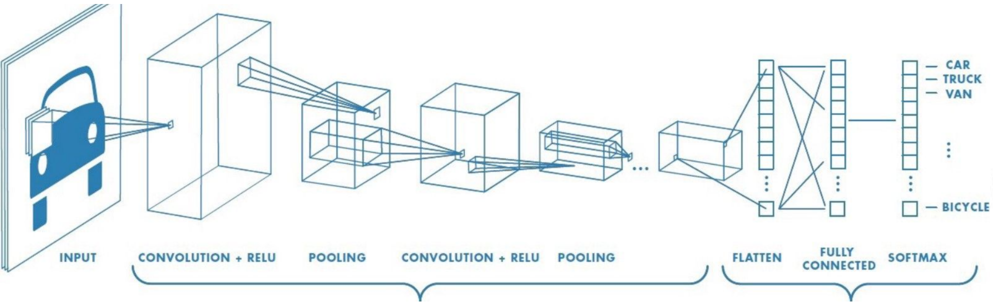


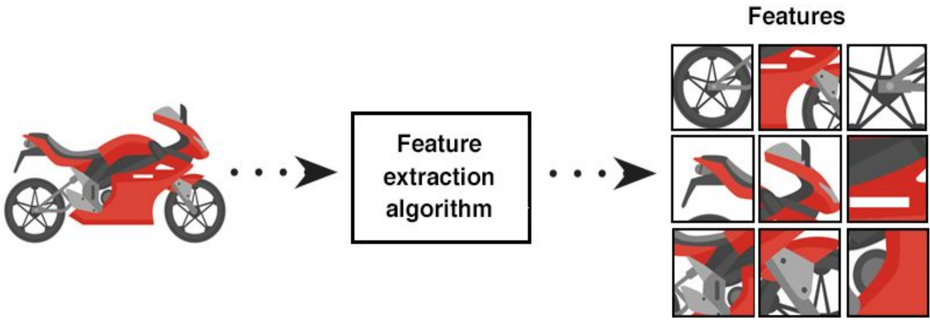


Computer Vision

Computer Vision is a field of Artificial Intelligence where the computer learns to derive some meaningful information from the given Image.

I got to learn what convolutions are and how do Convolutions work to figure out the features of the given image.





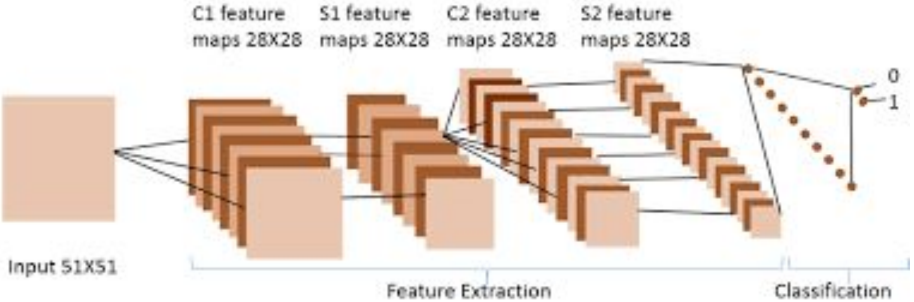
Crack Detection

Trained a pre-trained (on Imagenet Dataset) VGG model to classify image as cracked or uncracked.

Tried different Hyperparameters and found the best suited hyperparameter - Finetune a pre-trained model

Used tensorboard for visualisations.

Dataset split: Training - 11964 and Testing - 2991

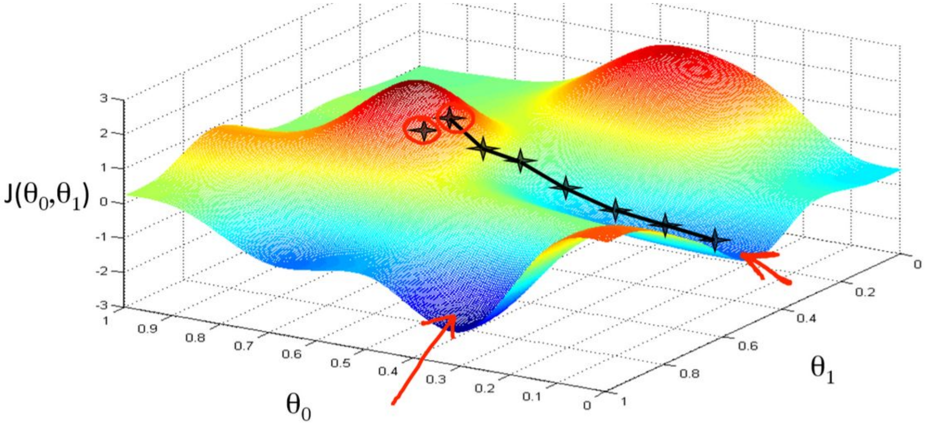


Cracked

Non-Cracked

Hyperparameters

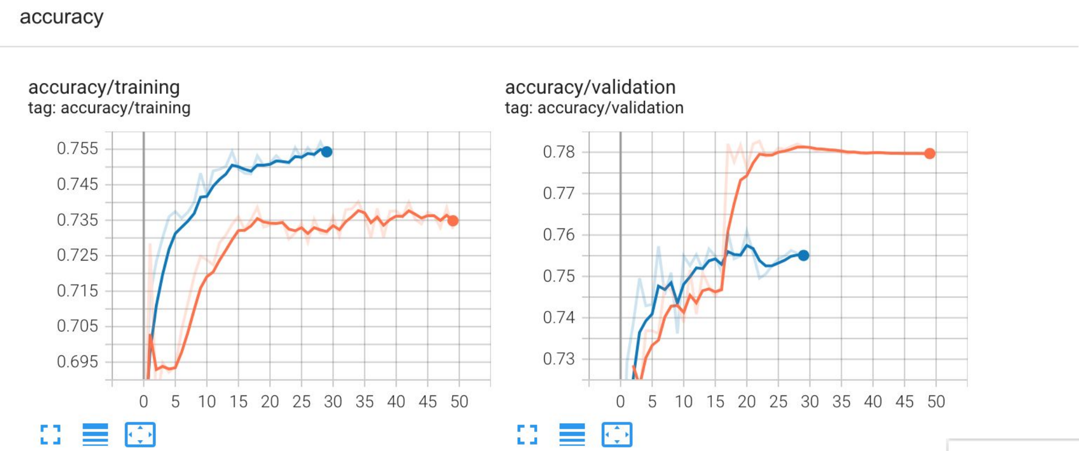
torch.optim

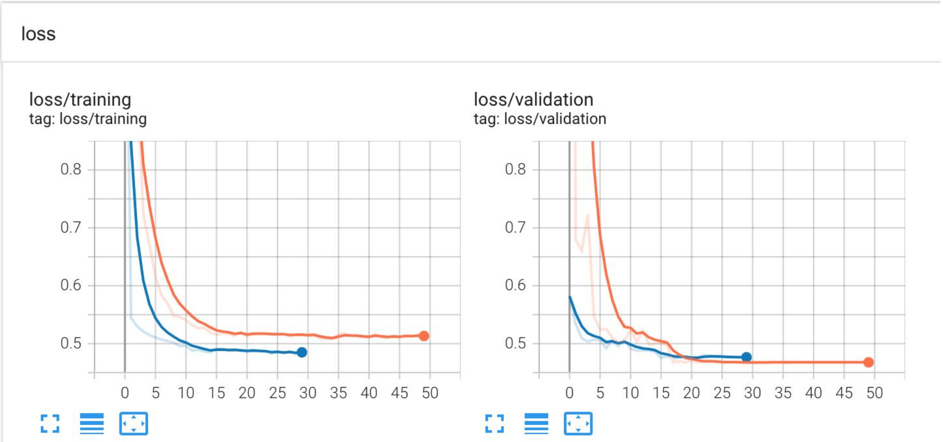


https://pytorch.org/docs/stable/optim.html

- Learning Rate
  - `lr_scheduler.StepLR`
  - `lr_scheduler.LinearLR`
  - `lr_scheduler.ExponentialLR`
  - `lr_scheduler.ReduceLROnPlateau`
- Momentum
- Optimizer Algorithms
  - `Adadelta`
  - `Adagrad`
  - `Adam`
  - `SGD`
- Batch Size
- Epochs

TensorBoard





- ● LR: 0.1
- ● LR: 0.01

Learnings

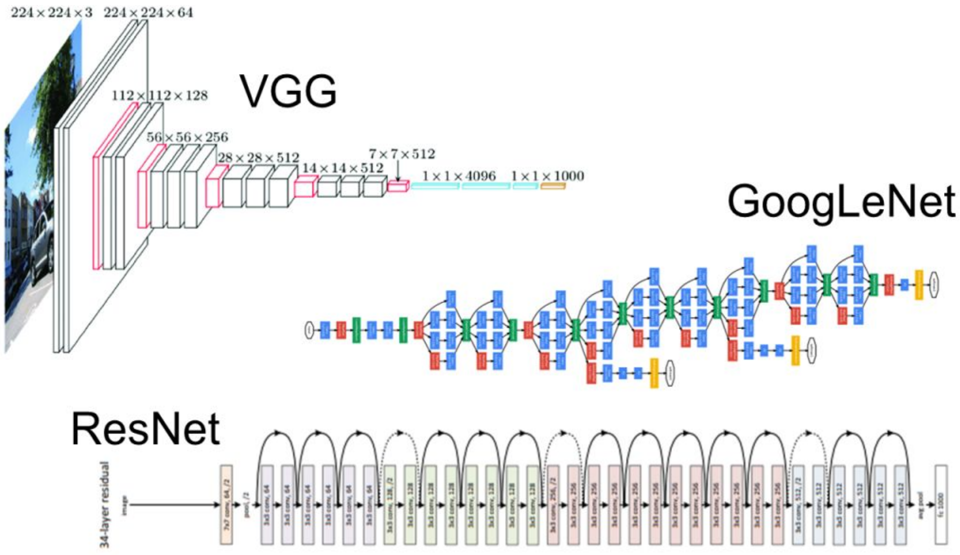
hit the button

- Learnt to debug the code
- Learnt how to find the best hyperparameters to train the model.
- Learnt how to write clean and optimized code.
- Going through Pytorch Documentations
- Learnt Tensorboard
- Optimized use of Colab

Challenges

- Colab has limited GPU and this was the biggest challenge while training the model.
- Significant results were only obtained when we run the model for more number of Epochs which becomes time consuming.

Future Possibilities



- We can train a newer and better model than VGG such as ResNet or DenseNet
- We can also create a custom model from scratch and train it to extract different features from images that are limited to crack detection
- We can also create a custom model that can detect scratches in phone screen and differentiate it from scratch in the screen guard.

CASHIFY